**PROJECT 2: Deploying Amazon RDS Multi-AZ and Read Replica, Simulate Failover**

**1.Creating an EC2 instance:**

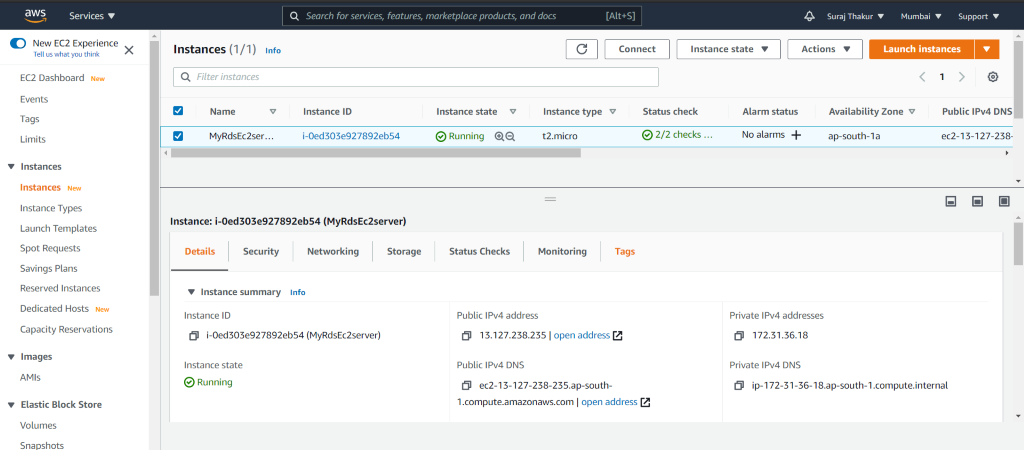
Step 1: Create an EC2 instance, Amazon Linux 2 AMI from the list of instances.  
Step 2: Configure the selected instance as follows:

a. Network: Select default available VPC   
 b. Subnet: Default selected   
 c. Auto-assign Public IP: Enable - It should be enabled because the public IP is needed for connecting to EC2 via SSH.   
 d. Under the User data section, enter the following script, (which installs MySQL): i. #!/bin/bash -ex  
 ii. yum install mysql -y

e. Add Storage Page: No need to change anything in this step

f. Add Tags Page a. Key: Name   
 b. Value: MyRdsEc2server   
 c. Click on  
 g. On the Configure Security Group page:   
 a. Assign a security group, Create a new security group   
 b. Security group name: MyEc2server-SG   
 c. Description: Security for ec2 server to connect with RDS   
 d. To add SSH, Source-Custom Enter 0.0.0.0/0 in the textbox or select Anywhere.  
 h. Review and Launch

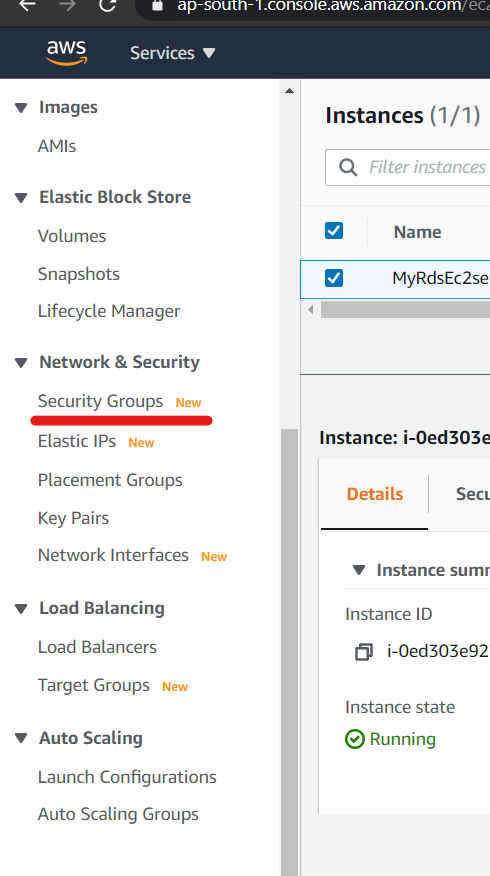
i. After 1-2 minutes, the Instance State will become running as shown below:



j. Note down IPv4 Private IP address in your text editor, navigate to the EC2 Dashboard and look in the instance details.

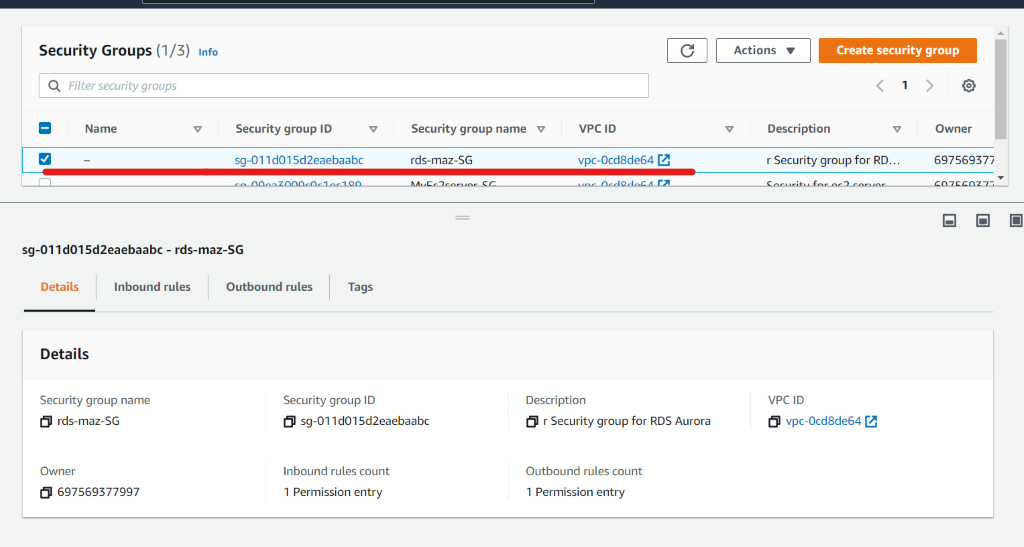
**2. Create a Security Group for RDS instance**  
  
**Step 1: Make sure that you are in the same region where you’ve created the ec2 instance.**

Step 2: Navigate to EC2, on the left menu panel select the security group.



Step 3: We are going to create a Security group for RDS with 3306 port number enabled.   
 a. Security group name: rds-maz-SG   
 b. Description: Enter Security group for RDS Aurora   
 c. VPC: Select Default VPC   
 d. Click on the button under Inbound rules.  
 i. In the textbox add 0.0.0.0/0   
 ii. Source: Select Custom   
 iii. Type: Select MYSQL/Aurora

Step 4: Leave everything as default and click on **Create**

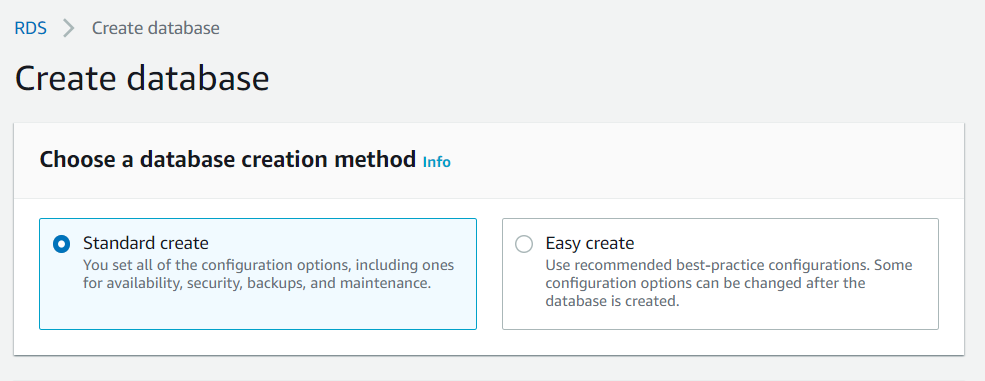


**3. Create an Amazon Aurora database with Multi-AZ enabled**  
  
**Step 1: Navigate to RDS under the database section of the Services menu**

Step 2: Click on Databases in the left Panel.

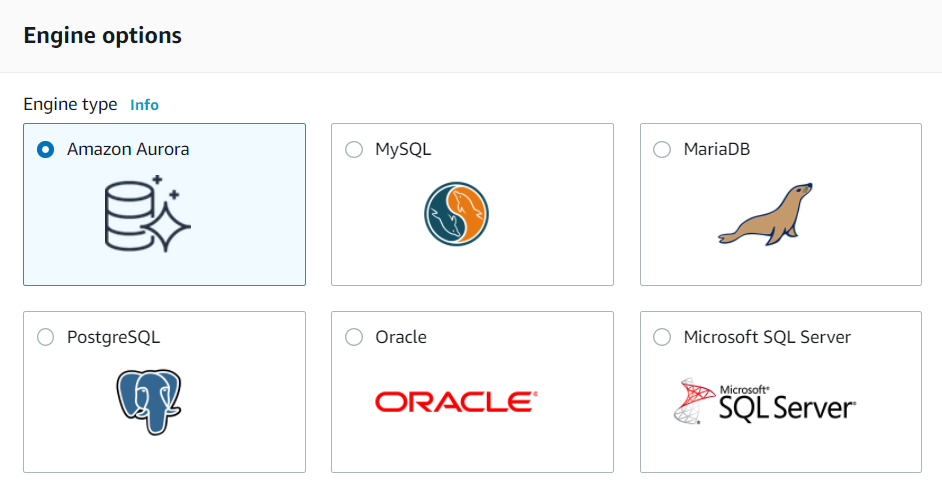
Step 3: Next we'll configure the database on the Create Database Page:

a. Choose a Database Creation Method:   
 i. Select Standard Create

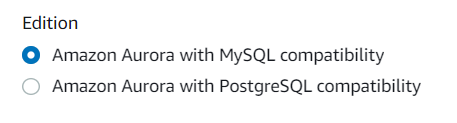


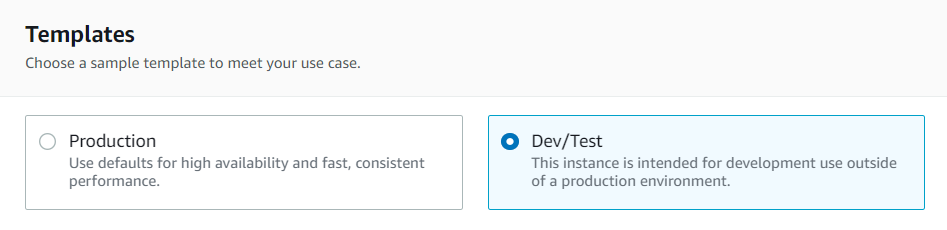
ii. In Engine options:

* Engine type: Choose Amazon Aurora

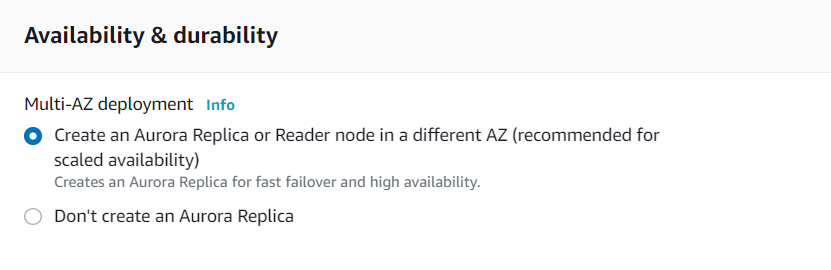


* Edition: Choose Amazon Aurora with MySQL compatibility

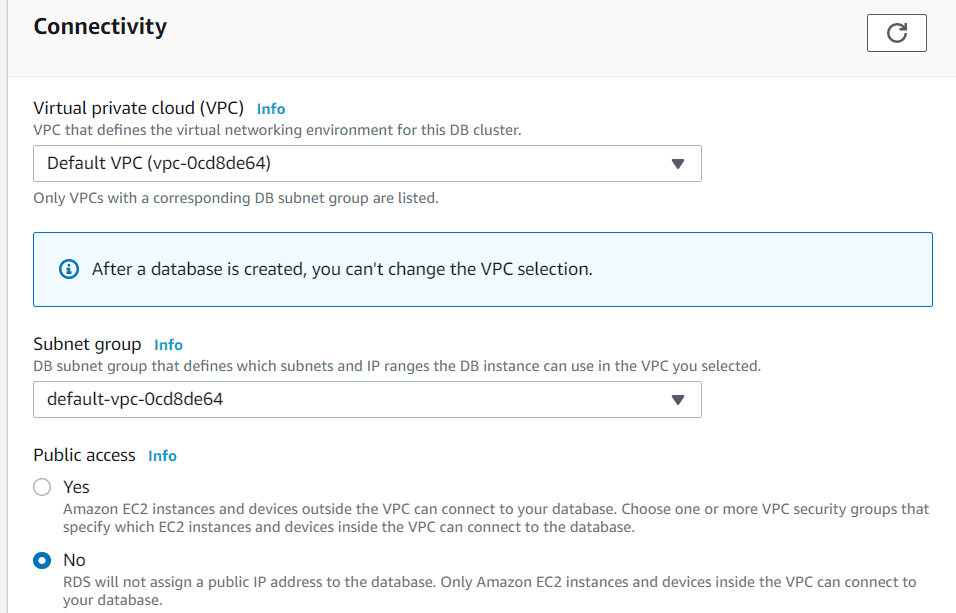


* Capacity Type: Choose Provisioned option for capacity type
* Version: Default (Aurora (MYSQL 5.7) 2.07.2)
* Replication features: Select Single-master (default)
* Choose Template: Dev/Test  
  
* Fill in the required details for the database (Aurora Cluster Settings)

DB cluster identifier: Specify cluster name: MyAuroraCluster

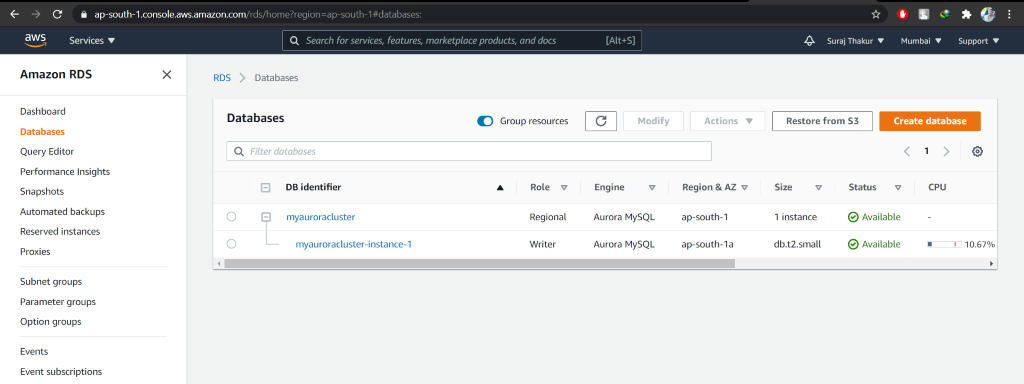
Give the following details in the credential settings:  
 1. Master Username: labsAdmin   
 2. Master password: labs12345   
 3. Confirm password: labs123   
 4.Note: This is the username and password used to log into your database. Please make note of them.   
 5. Choosing DB instance size   
 6. DB instance class: Choose Burstable classes (includes t classes)   
 7. By Default, db.t3.small will be selected, please enable the option below of Include previous generation classes.   
 8. Select db.t2.small instance.  
Note: Incase, db.t2.small is not available, we have added support for db.t3.small also, meaning you can go ahead with any of the two types.  
  
 Availability and Durability: Choose Multi-AZ deployment: Create Aurora Replica or Reader node in a different AZ as shown below:  
 

* Connectivity: Choose the Default VPC
* Additional connectivity configuration   
  i. Subnet group: Default   
  ii. Publicly accessible: Yes   
  iii. Existing VPC security groups



* Select rds-maz-SG for the dropdown. (This is the security group which you have created in the beginning)   
  i. Database port: 3306
* Additional configuration:  
  Initial database name: whizlabsrds   
  i. Encryption: Uncheck   
  ii. Leave other settings as default

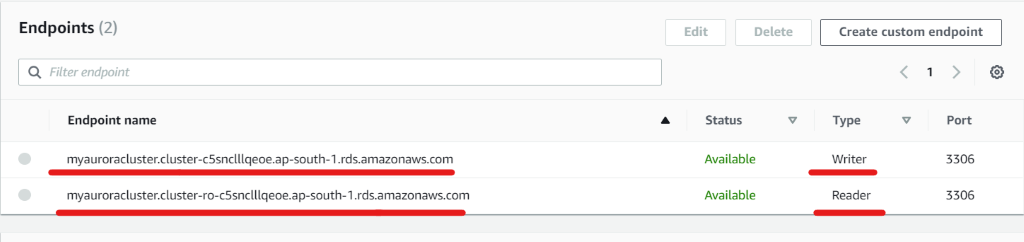
Step 4: After completion of filling all the configuration, it will take around 10-15 minutes for the database to be created. Please wait until database status changes from creating to available.  
  
Step 5: Once the database has been created, you should see the following page:

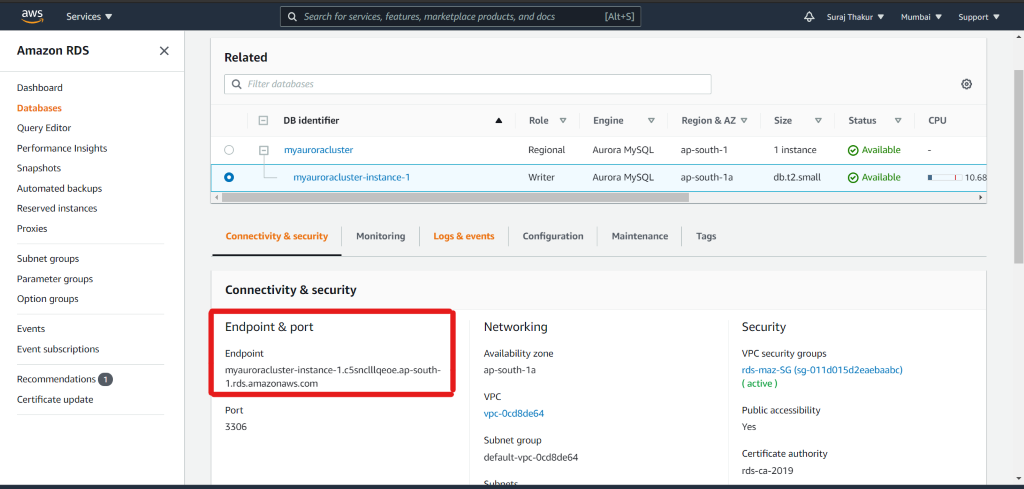


Similar to the screenshot, you should be able to see that our database launched in multiple AZs, namely ap-south-1 and ap-south-1a

**4. Aurora (MySQL) database on Connecting to the RDS**

* After launching Aurora RDS with multi-AZ enabled successfully. To connect to RDS database, we need the end point.  
  a. Click on the RDS cluster name and then navigate to Connectivity & security to find the endpoint of your Master (Writer) and Reader instances, with which you can connect to your DB instance.
* The endpoints you see to be similar to these examples:   
  i. Master (Writer): myauroracluster.cluster-cpoz6c7903cx.us-east-1.rds.amazonaws.com   
  ii. Reader: myauroracluster.cluster-ro-cpoz6c7903cx.us-east-  
  1.rds.amazonaws.



To get the endpoint of the RDS instances, click on the name of the cluster. Then you should click on Endpoints. This will expose the read and write endpoints for the database. See the example below:  


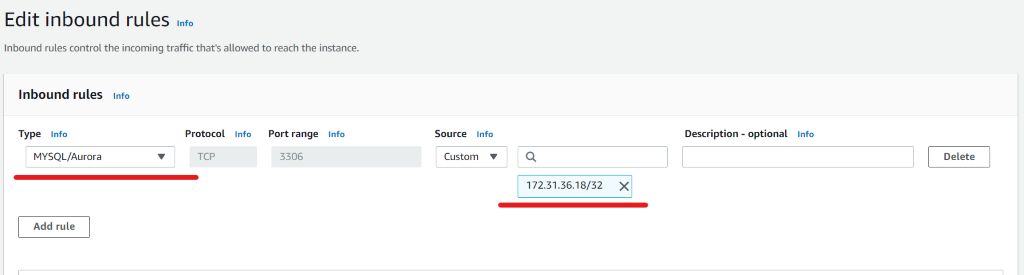
**5. Connecting the EC2 Server to RDS**

a. Now we need to connect the RDS with ec2 server in order to eventually connect with the Aurora database.

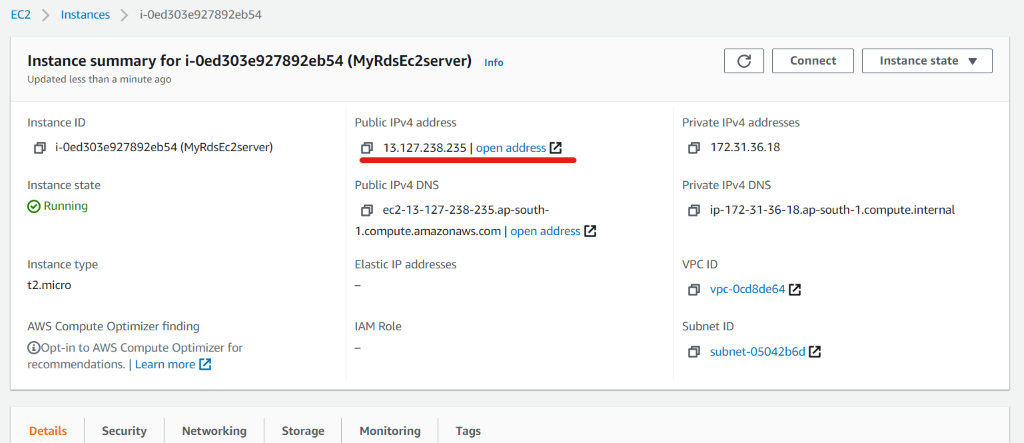
b. Navigate to RDS available under the database section of the Services menu.

c. Click on Master (writer) database and click on the security group name in this example it is rds-maz-SG under VPC security groups.  
d. It will open the Security Group page. Click on InBound.

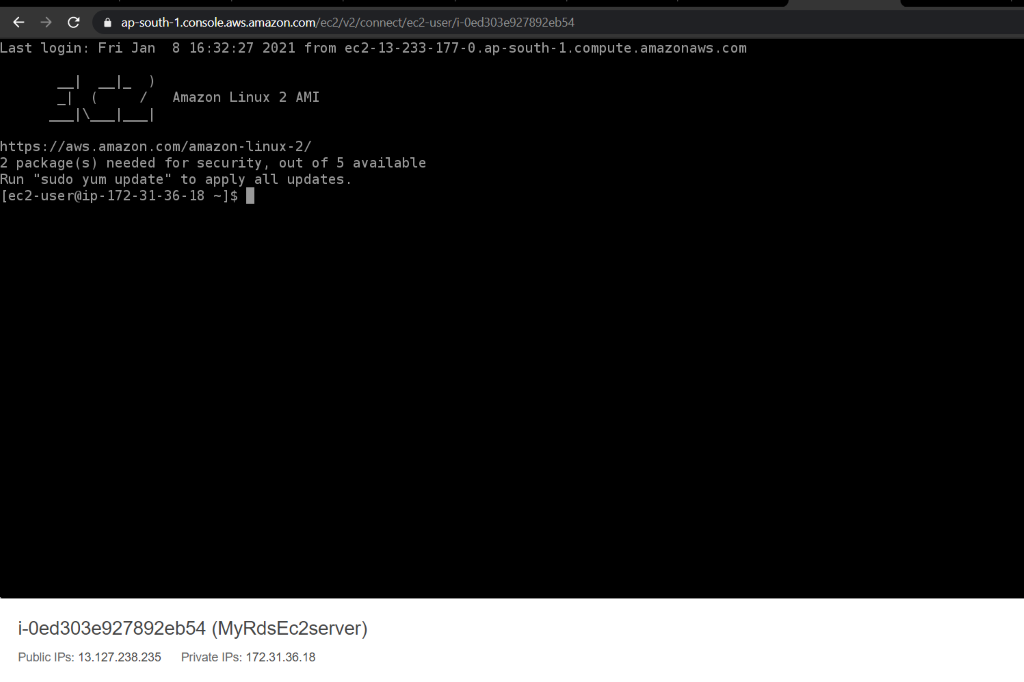
i. The MySQL rule will already exists.  
 ii. Under source, delete any pre-populated IP Address and enter the private IP of your MyRdsEc2server EC2 instance with CIDR /32 (EC2 instance Private IP) and then click as shown below:



**6. Execute Database Operations via SSH**

a. Copy the IPv4 Public IP address, navigate to the EC2 Dashboard and look in the instance details.  


b. SSH into the EC2 instance we just created through the following steps in SSH into EC2 Instance.



c. Switch to the root user using the command: sudo –s

i. Log into the RDS instance using the below command:

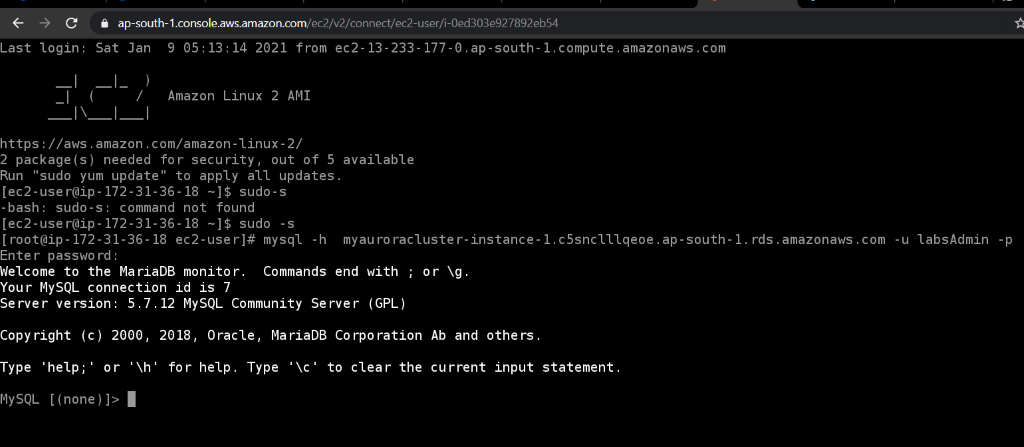
Syntax: **mysql -h <endpoint name> -u <username> -p**

Make sure to change the above Master (Writer)Cluster endpoint and Username with your's .

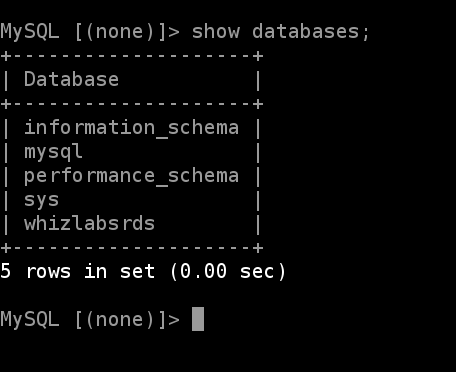
Host name: myauroracluster.cluster-cpoz6c7903cx.us-east-1.rds.amazonaws.com

Username: labsAdmin

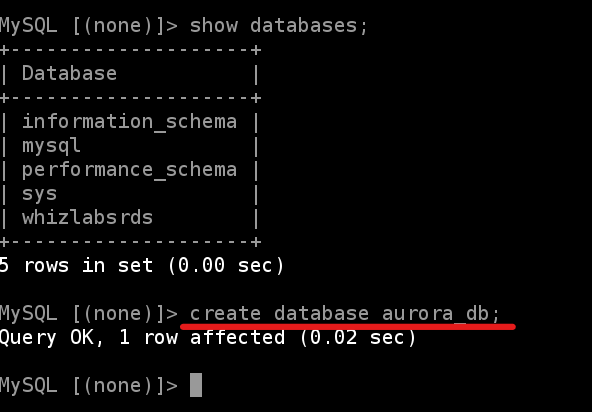
Password: labs123 (Use yours in case you changed the password while creating RDS)



ii. List all Databases using the following command: show databases;



iii. We'll create a demo database named aurora\_db :   
 **CREATE DATABASE AURORA\_DB;**

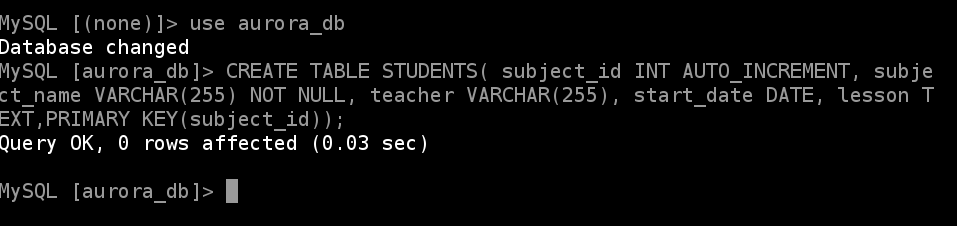


Iv. Select the newly-created database:

**USE AURORA\_DB;**

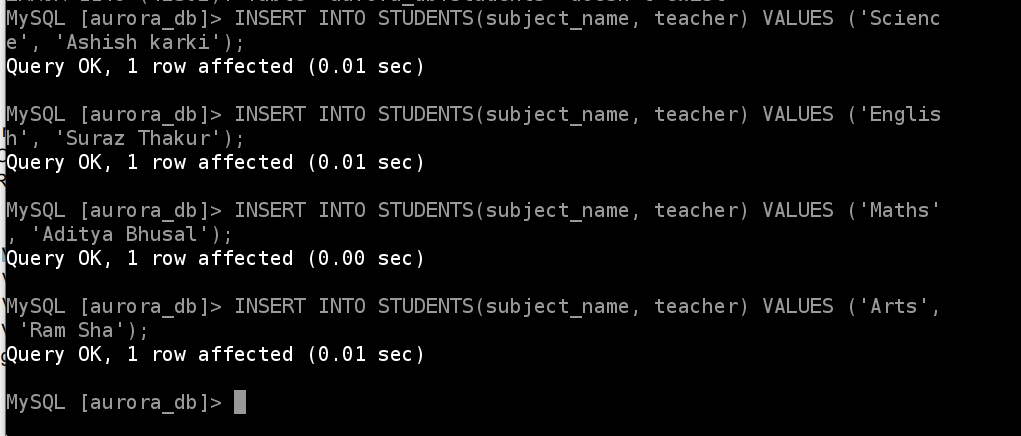
v. Next, we'll create a table named students and insert few rows of data using list of commands:

**CREATE TABLE students ( subject\_id INT AUTO\_INCREMENT, subject\_name VARCHAR (255) NOT NULL, teacher VARCHAR (255),start\_date DATE, lesson TEXT, PRIMARY KEY (subject\_id));**



vi. Insert data into the table:

a. **INSERT INTO students(subject\_name, teacher) VALUES ('English', 'John Taylor');**   
b. **INSERT INTO students(subject\_name, teacher) VALUES ('Science', 'Mary Smith');**   
c. **INSERT INTO students(subject\_name, teacher) VALUES ('Maths', 'Ted Miller');**   
d. **INSERT INTO students(subject\_name, teacher) VALUES ('Arts', 'Suzan Carpenter');**



vii. Now you can view the contents of the table student using the below command: **SELECT \* from students;**

viii. Exit from mysql console use the below command: **EXIT**

**7. Forcing a Failover to Test Multi-AZ**

a. To test if Multi-AZ is working, we will create a situation where master fails and the read replica has to become the new Master (Writer).   
  
b. On the next screen, confirm the Failover.   
c. Wait for a few minutes for the RDS instances to failover. a. (i.e., Master (Writer) becomes Reader and Reader becomes Master (Writer))

**8. Testing the Failover Condition**  
  
a. Now connect to RDS with new Master endpoint, Copy the endpoint of the new Master (Writer)cluster and replace it with your endpoint link.   
 **Mysql -h <endpoint>-u <username> -p**  
b. Then it will ask for the password, provide the password and hit enter.

c. You will be able to Log into MySQL and check for the database and table created in the master DB instance before the failover.

You can notice the resources created on the original master db are present, implying that the Failover worked successfully.

d. Use the following commands for checking the databases after successful failover:

**Show databases;**

**USE aurora\_db**

**e.** Now check the existence of table named students and data (that we created earlier in the lab):

**SHOW tables;**

**SELECT \* from students;**

